Computer Graphics (UCS505)

Project on

Satellite Animation

Submitted By

Aashima	101917113
Harshita Gupta	101917125
Jaskaran Singh Purewal	101917129

B.E. Third Year – CSE5

Group No. -- 8

Submitted To:

Dr. Samya Muhuri

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology

Patiala – 147001

Table of Contents

INTRODUCTION:

Satellite Animation is a simulation of take-off of the rocket which will deploy the satellite, deployment of the satellite and the subsequent orientation correction of its orbit around a planet. This graphics package is based on the OpenGL library functions. The programming language used here is C++ using OpenGL libraries.

The aim of this project is to provide an adept graphical visualization of deploying a satellite via a rocket, and its revolution around a planet about its set orbit.

You have 2 options in the menu at the title page:

By pressing S – key: Launch the satellite. (Start the animation)

By pressing Q – key: Quit the animation.

CONCEPTS:

Graphics provides one of the most natural means of communicating with a computer, since our highly developed 2D Or 3D pattern-recognition abilities allow us to perceive and process pictorial data rapidly. Interactive graphics is the most important means of producing pictures since the invention of photography and television. We can make pictures of not only the real world objects but also of abstract objects such as mathematical surfaces on 4D and of data that have no inherent geometry.

Translation Function: A translation process moves every point a constant distance in a specified direction. It can be described as a rigid motion. A translation can also be interpreted as the addition of a constant vector to every point, or as shifting the origin of the coordinate system.

Scaling: It is used to alter or change the size of objects. The change is done using scaling factors. There are two scaling factors, i.e. Sx in x direction Sy in y-direction. If the original position is x and y. Scaling factors are Sx and Sy then the value of coordinates after scaling will be x1 and y1.

Animation: It is a method, where objects are manipulated to appear as moving, without the user input. It requires a form of infinite loop, which is usually achieved by a set of functions or class methods that describe the changes to each object in a small unit of time- called update or move.

Polygon : By specifying the vertices we have made different polygon , using GL_POLYGON from the GL/glut library. Lines: By using GL_LINES we have drawn lines in our project.

RGB Algorithm: The RGB Color Model is used for color representation, it is a color coordinate system having three primary colors, each primary color having its intensity value ranging from 0 to 1. Mixing these three primary colors at varying intensities produces a variety of colors.

USER-DEFINED FUNCTIONS:

1. void drawFilledCircle(GLfloat x, GLfloat y, GLfloat radius)

   This function is used to draw a filled circle.

2. void drawstring(int x, int y, char *s)

   This function will draw an input string s on the screen.

3. void semicircle(float radius,float u,float v)

   This function draws a semicircle of given radius and center on the screen.

4. void control()

   Determines the state of Satellite launch.

5. void stars()

   Twinkling stars animation.

6. void stars1()

   Different twinkling stars animation.

7. void static_rocket()

   Scene of the Rocket on ground.

8. void rocket_to_cam_pos()

   Rocket animation in the atmosphere.

9. void rocket_in_motion()

   Satellite animation in space.

10. void mars(float radius)

    Planet animation. (Mars)

11. void keyboard(unsigned char key, int x, int y)

    Manage keyboard inputs.

12. void page()

    Design of Title page.

13. void display()

    Display all components.

14. void myinit()

    Initialize window colors, mode, and point size.

CODE:

main.cpp

```
#include<GL/glut.h>
#include<stdlib.h>
#include<stdio.h>
#include<math.h>
#include<string.h>
const float DEG2RAD = 3.14159/180;
void stars();
int p;
void stars1();
void static_rocket();
void rocket_to_cam_pos();
void rocket_in_motion();
void mars(float radius);
float tx=0;
float xx=1;
float yy=1;
float i,j,count=0,count1=0,count3=0,flag=0,flag1=0,t=0,f=0,flag3=0;

// fucntion to display the text content of the home screen
void drawFilledCircle(GLfloat x, GLfloat y, GLfloat radius){
    int i;
    int triangleAmount = 20; //# of triangles used to draw circle


    GLfloat twicePi = 2.0f * 3.14;

    glBegin(GL_TRIANGLE_FAN);
        glVertex2f(x, y); // center of circle
        for(i = 0; i <= triangleAmount;i++) {
            glVertex2f(
                    x + (radius * cos(i *  twicePi / triangleAmount)),
                y + (radius * sin(i * twicePi / triangleAmount))
            );
        }
    glEnd();
}
void drawstring(int x, int y, char *s)
{
    char *c;
    glRasterPos2i(x, y);
    for (c = s; *c != '\0'; *c++)
        glutBitmapCharacter(GLUT_BITMAP_8_BY_13, *c);
}

```

```
43  }
44
45  void semicircle(float radius,float u,float v)
46  {
47
48      glColor3f(1.0 ,1.0 ,1.0);
49     glBegin(GL_POLYGON);
50
51     for (int i=135; i<=315; i++)
52     {
53        float degInRad = i*DEG2RAD;
54        glVertex2f(u+cos(degInRad)*radius,v+(sin(degInRad))*radius);//100,100 specifies centre
55     }
56
57     glEnd();
58  }
59
60  //determines the state of rocket launch
61  void control()
62  {
63      count1++;
64      if(count1==25000)
65              flag=1;
66
67      else if (flag == 1 && (count1 == 60000))
68          rocket_to_cam_pos();
69
70      else if (flag == 1 && count1 >= 100000)
71          rocket_in_motion();
72  }
73
74  void stars()
75  {
76
77      glColor3f(1.0,1.0,1.0);
78      glPointSize(1.37);
79      glBegin(GL_POINTS);
80      glVertex2i(10,20);
81      glVertex2i(20,100);
82      glVertex2i(30,10);
83      glVertex2i(15,150);
84      glVertex2i(17,80);
85      glVertex2i(200,200);
86      glVertex2i(55,33);
```

```
85      glVertex2i(200,200);
86      glVertex2i(55,33);
87      glVertex2i(400,300);
88      glVertex2i(330,110);
89      glVertex2i(125,63);
90      glVertex2i(63,125);
91      glVertex2i(20,10);
92      glVertex2i(110,330);
93      glVertex2i(440,430);
94      glVertex2i(32,65);
95      glVertex2i(110,440);
96      glVertex2i(210,230);
97      glVertex2i(390,490);
98      glVertex2i(12,90);
99      glVertex2i(400,322);
100     glVertex2i(420,366);
101     glVertex2i(455,400);
102     glVertex2i(20,20);
103     glVertex2i(111,120);
104     glVertex2i(401,200);
105     glVertex2i(230,30);
106     glVertex2i(220,20);
107     glVertex2i(122,378);
108     glVertex2i(133,340);
109     glVertex2i(345,420);
110     glVertex2i(130,360);
111     glVertex2i(333,120);
112     glVertex2i(250,22);
113     glVertex2i(242,11);
114     glVertex2i(280,332);
115     glVertex2i(233,40);
116     glVertex2i(210,418);
117     glVertex2i(256,12);
118     glVertex2i(288,232);
119     glVertex2i(247,36);
120     glVertex2i(229,342);
121     glVertex2i(257,47);
122     glVertex2i(290,63);
123     glVertex2i(232,72);
124     glVertex2i(243,143);
125     glVertex2i(100,200);
126     glVertex2i(90,250);
127     glVertex2i(80,225);
128     glVertex2i(50,333);
```

```
127         glVertex2i(80,225);
128         glVertex2i(50,333);
129         glVertex2i(60,350);
130         glVertex2i(243,143);
131         glVertex2i(243,143);
132         glEnd();
133     }
134
135     void starsl()
136     {
137         int l;
138         glColor3f(1.0,1.0,1.0);
139         glPointSize(1.0);
140         glBegin(GL_POINTS);
141         glVertex2i(50,20);
142         glVertex2i(70,100);
143         glVertex2i(80,10);
144         glVertex2i(65,150);
145         glVertex2i(67,80);
146         glVertex2i(105,33);
147         glVertex2i(450,300);
148         glVertex2i(380,110);
149         glVertex2i(175,63);
150         glVertex2i(113,125);
151         glVertex2i(70,10);
152         glVertex2i(160,330);
153         glVertex2i(490,430);
154         glVertex2i(82,65);
155         glVertex2i(160,440);
156         glVertex2i(440,490);
157         glVertex2i(62,90);
158         glVertex2i(450,322);
159         glVertex2i(420,366);
160         glVertex2i(455,400);
161         glVertex2i(60,20);
162         glVertex2i(111,120);
163         glVertex2i(451,200);
164         glVertex2i(280,30);
165         glVertex2i(220,20);
166         glVertex2i(132,378);
167         glVertex2i(173,340);
168         glVertex2i(325,420);
169         glVertex2i(180,360);
170         glVertex2i(383,120);
```

```
169         glVertex2i(180,360);
170         glVertex2i(383,120);
171         glVertex2i(200,22);
172         glVertex2i(342,11);
173         glVertex2i(330,332);
174         glVertex2i(283,40);
175         glVertex2i(210,418);
176         glVertex2i(256,12);
177         glVertex2i(288,232);
178         glVertex2i(247,36);
179         glVertex2i(229,342);
180         glVertex2i(257,47);
181         glVertex2i(290,63);
182         glVertex2i(232,72);
183         glVertex2i(243,143);
184         glVertex2i(100,200);
185         glVertex2i(90,250);
186         glVertex2i(80,225);
187         glVertex2i(50,333);
188         glVertex2i(60,350);
189         glVertex2i(243,143);
190         glVertex2i(243,143);
191         glEnd();
192         for(l=0;l<=10000;l++)
193             ;
194     }
195     void static_rocket()
196     {
197             glClearColor(0.196078  ,0.6 ,0.8,1.0);
198         glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
199
200         glColor3f(0.133,0.545,0.133);
201             glBegin(GL_POLYGON);//green ground
202             glVertex2f(0.0,0.0);
203             glVertex2f(0.0,250.0);
204             glVertex2f(270.0,250.0);
205             glVertex2f(500.0,50.0);
206             glVertex2f(500.0,0.0);
207             glEnd();
208             glBegin(GL_POLYGON);//green ground
209             glVertex2f(280.0,250.0);
210             glVertex2f(500.0,250.0);
211             glVertex2f(500.0,60.0);
212             glEnd();
```

```
211         glVertex2f(500.0,60.0);
212         glEnd();
213         glColor3f(0.0,0.0,0.0);
214             glBegin(GL_POLYGON);//road
215         glVertex2f(260.0,250.0);
216         glVertex2f(290.0,250.0);
217         glVertex2f(500.0,70.0);
218         glVertex2f(500.0,40.0);
219         glEnd();
220         glColor3f(0.0,0.0,0.0);
221
222
223         glColor3f(0.8,0.498039 ,0.196078);
224             glBegin(GL_POLYGON);//house 1
225         glVertex2f(250.0,250.0);
226         glVertex2f(300.0,250.0);
227         glVertex2f(300.0,350.0);
228         glVertex2f(250.0,350.0);
229         glEnd();
230         glColor3f(0.7,0.7,0.7);
231         glBegin(GL_POLYGON);//HOUSE A
232             glVertex2f(255,267.5);
233             glVertex2f(275.0,267.5);
234             glVertex2f(275.0,277.5);
235             glVertex2f(255.0,277.5);
236             glEnd();
237         glBegin(GL_POLYGON);//HOUSE B
238             glVertex2f(255,285.0);
239             glVertex2f(275.0,285);
240             glVertex2f(275.0,295);
241             glVertex2f(255.0,295);
242             glEnd();
243
244         glBegin(GL_POLYGON);//HOUSE C
245             glVertex2f(255,302.5);
246             glVertex2f(275.0,302.5);
247             glVertex2f(275.0,312.5);
248             glVertex2f(255.0,312.5);
249             glEnd();
250
251         glBegin(GL_POLYGON);//HOUSE D
252             glVertex2f(255,320.0);
253             glVertex2f(275.0,320.0);
254             glVertex2f(275.0,330.0);
```

```
253             glVertex2f(275.0,320.0);
254             glVertex2f(275.0,330.0);
255             glVertex2f(255.0,330.0);
256             glEnd();
257
258         glBegin(GL_POLYGON);//HOUSE E
259             glVertex2f(285,267.5);
260             glVertex2f(295.0,267.5);
261             glVertex2f(295.0,277.5);
262             glVertex2f(285.0,277.5);
263             glEnd();
264
265         glBegin(GL_POLYGON);//HOUSE F
266             glVertex2f(285,285.0);
267             glVertex2f(295.0,285);
268             glVertex2f(295.0,295);
269             glVertex2f(285.0,295);
270             glEnd();
271
272         glBegin(GL_POLYGON);//HOUSE G
273             glVertex2f(285,302.5);
274             glVertex2f(295.0,302.5);
275             glVertex2f(295.0,312.5);
276             glVertex2f(285.0,312.5);
277             glEnd();
278
279         glBegin(GL_POLYGON);//HOUSE H
280             glVertex2f(285,320.0);
281             glVertex2f(295.0,320.0);
282             glVertex2f(295.0,330.0);
283             glVertex2f(285.0,330.0);
284             glEnd();
285             glColor3f(0.647059 ,0.164706  ,0.164706);
286             glBegin(GL_POLYGON);//solid cone
287             glVertex2f(26,250);
288             glVertex2f(52,250);
289             glVertex2f(39,290);
290             glEnd();
291             semicircle(20.0,50,300);
292
293
294     glColor3f(1.0,1.0 ,1.0);
295             glBegin(GL_POINTS);//road paint
296             glVertex2f(497,56);
```

```
295             glBegin(GL_POINTS);//road paint
296             glVertex2f(497,56);
297             glVertex2f(488,65);
298             glVertex2f(479,74);
299             glVertex2f(470,83);
300             glVertex2f(460,92);
301             glVertex2f(450,101);
302             glVertex2f(439,110);
303             glVertex2f(428,119);
304             glVertex2f(418,128);
305             glVertex2f(408,137);
306             glVertex2f(398,146);
307             glVertex2f(388,155);
308             glVertex2f(378,164);
309             glVertex2f(366,173);
310             glVertex2f(356,182);
311             glVertex2f(346,191);
312             glVertex2f(336,200);
313             glVertex2f(324,209);
314             glVertex2f(314,218);
315             glVertex2f(304,227);
316             glVertex2f(294,234);
317             glVertex2f(284,243);
318             glVertex2f(278,248);
319
320             glEnd();
321
322
323         glColor3f(0.0,0.0,0.0);//stand object
324         glBegin(GL_POLYGON);
325         glVertex2f(130,10.0);
326         glVertex2f(160,10.0);
327         glVertex2f(160,180.0);
328         glVertex2f(130,180.0);
329         glEnd();
330         glBegin(GL_LINES);
331         glVertex2f(130,30.0);
332         glVertex2f(262,30.0);
333
334         glVertex2f(130,130.0);
335         glVertex2f(260,130.0);
336         glEnd();
337
338         glColor3f(0.8,0.498039 ,0.196078);
```

```
337
338         glColor3f(0.8,0.498039 ,0.196078);
339         glBegin(GL_POLYGON);//core
340             glVertex2f(237.5,20.0);
341             glVertex2f(262.5,20.0);
342             glVertex2f(262.5,120.0);
343             glVertex2f(237.5,120.0);
344         glEnd();
345
346         glColor3f(1.0,1.0,1.0);//bonnet
347         glBegin(GL_POLYGON);//front
348         glVertex2f(237.5,120.0);
349         glVertex2f(262.5,120.0);
350         glVertex2f(250,170.0);
351         glEnd();
352         glColor3f(1.0,0.0,0.0);
353         glBegin(GL_POLYGON);//left_side_top
354         glVertex2f(237.5,120.0);
355         glVertex2f(217.5,95.0);
356         glVertex2f(237.5,95.0);
357         glEnd();
358             glBegin(GL_POLYGON);//left_side_bottom
359         glVertex2f(237.5,20.0);
360         glVertex2f(217.5,20.0);
361         glVertex2f(237.5,70.0);
362         glEnd();
363             glBegin(GL_POLYGON);//right_side_bottom
364         glVertex2f(262.5,20.0);
365         glVertex2f(282.5,20.0);
366         glVertex2f(262.5,70.0);
367         glEnd();
368             glBegin(GL_POLYGON);//right_side_top
369         glVertex2f(262.5,120.0);
370         glVertex2f(262.5,95.0);
371         glVertex2f(282.5,95.0);
372         glEnd();
373         glColor3f(0.556863 ,0.137255  ,0.419608);
374             glBegin(GL_POLYGON);//bottom_1_exhaust
375         glVertex2f(237.5,20.0);
376         glVertex2f(244.5,20.0);
377         glVertex2f(241,0.0);
378         glEnd();
379             glBegin(GL_POLYGON);//bottom_2_exhaust
380         glVertex2f(246.5,20.0);
```

```
379         glBegin(GL_POLYGON);//bottom_2_exhaust
380     glVertex2f(246.5,20.0);
381     glVertex2f(253.5,20.0);
382     glVertex2f(249.5,0.0);
383     glEnd();
384         glBegin(GL_POLYGON);//bottom_3_exhaust
385     glVertex2f(262.5,20.0);
386     glVertex2f(255.5,20.0);
387     glVertex2f(258.5,0.0);
388     glEnd();
389
390     glBegin(GL_POLYGON);//left_stand_holder
391     glVertex2f(182.5,85.0);
392     glVertex2f(182.5,0.0);
393     glVertex2f(187.5,0.0);
394     glVertex2f(187.5,80.0);
395     glVertex2f(237.5,80.0);
396     glVertex2f(237.5,85.0);
397     glVertex2f(182.5,85.0);
398     glEnd();
399     glBegin(GL_POLYGON);
400     glVertex2f(312.5,85.0);//right_stand_holder
401     glVertex2f(312.5,0.0);
402     glVertex2f(307.5,0.0);
403     glVertex2f(307.5,80.0);
404     glVertex2f(262.5,80.0);
405     glVertex2f(262.5,85.0);
406     glVertex2f(312.5,85.0);
407     glEnd();
408     glColor3f(0,0,1);
409     drawstring(260,350,"TESLA");
410     glutSwapBuffers();
411     glutPostRedisplay();
412     glFlush();
413
414
415 }
416 void rocket_to_cam_pos()
417 {
418     count++;
419 count3++;
420
421 for(float i=0;i<=500;i+=0.3333)
422 {
```

```
for(float i=0;i<=500;i+=0.3333)
{



    glClearColor(0.196078  ,0.6 ,0.8,1.0);
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);

    glColor3f(0.8,0.498039 ,0.196078);
    glBegin(GL_POLYGON);//core
        glVertex2f(237.5,20.0+i);
        glVertex2f(262.5,20.0+i);
        glVertex2f(262.5,120.0+i);
        glVertex2f(237.5,120.0+i);


    glEnd();

    glColor3f(1.0,1.0,1.0);//bonnet
    glBegin(GL_POLYGON);//front
    glVertex2f(237.5,120.0+i);
    glVertex2f(262.5,120.0+i);
    glVertex2f(250,170.0+i);
    glEnd();
    glColor3f(1.0,0.0,0.0);
    glBegin(GL_POLYGON);//left_side_top
    glVertex2f(237.5,120.0+i);
    glVertex2f(217.5,95.0+i);
    glVertex2f(237.5,95.0+i);
    glEnd();
        glBegin(GL_POLYGON);//left_side_bottom
    glVertex2f(237.5,20.0+i);
    glVertex2f(217.5,20.0+i);
    glVertex2f(237.5,70.0+i);
    glEnd();
        glBegin(GL_POLYGON);//right_side_bottom
    glVertex2f(262.5,20.0+i);
    glVertex2f(282.5,20.0+i);
    glVertex2f(262.5,70.0+i);
    glEnd();
        glBegin(GL_POLYGON);//right_side_top
    glVertex2f(262.5,120.0+i);
    glVertex2f(262.5,95.0+i);
    glVertex2f(282.5,95.0+i);
    glEnd();
```

```
463         glVertex2f(282.5,95.0+i);
464         glEnd();
465         glColor3f(0.556863 ,0.137255  ,0.419608);
466             glBegin(GL_POLYGON);//bottom_1_exhaust
467         glVertex2f(237.5,20.0+i);
468         glVertex2f(244.5,20.0+i);
469         glVertex2f(241,0.0+i);
470         glEnd();
471             glBegin(GL_POLYGON);//bottom_2_exhaust
472         glVertex2f(246.5,20.0+i);
473         glVertex2f(253.5,20.0+i);
474         glVertex2f(249.5,0.0+i);
475         glEnd();
476             glBegin(GL_POLYGON);//bottom_3_exhaust
477         glVertex2f(262.5,20.0+i);
478         glVertex2f(255.5,20.0+i);
479         glVertex2f(258.5,0.0+i);
480         glEnd();
481
482         if((p%2)==0)
483                     glColor3f(1.0,0.25,0.0);
484                     else
485                         glColor3f(1.0,0.816,0.0);
486
487                     glBegin(GL_POLYGON);//outer fume
488             glVertex2f(237.5,20+i);
489             glVertex2f(234.16,16.66+i);
490             glVertex2f(230.82,13.32+i);
491             glVertex2f(227.48,9.98+i);
492             glVertex2f(224.14,6.64+i);
493             glVertex2f(220.8,3.3+i);
494             glVertex2f(217.5,0+i);
495             glVertex2f(221.56,-5+i);
496             glVertex2f(225.62,-10+i);
497             glVertex2f(229.68,-15+i);
498             glVertex2f(233.74,-20+i);
499             glVertex2f(237.8,-25+i);
500             glVertex2f(241.86,-30+i);
501             glVertex2f(245.92,-35+i);
502             glVertex2f(250,-40+i);
503             glVertex2f(254.06,-35+i);
504             glVertex2f(258.12,-30+i);
505             glVertex2f(262.18,-25+i);
506             glVertex2f(266.24,-20+i);
```

```
505         glVertex2f(262.18,-25+i);
506         glVertex2f(266.24,-20+i);
507         glVertex2f(270.3,-15+i);
508         glVertex2f(274.36,-10+i);
509         glVertex2f(278.42,-5+i);
510         glVertex2f(282.5,0+i);
511         glVertex2f(278.5,4+i);
512         glVertex2f(274.5,8+i);
513         glVertex2f(270.5,12+i);
514         glVertex2f(266.5,16+i);
515         glVertex2f(262.5,20+i);//28 points
516         glEnd();
517
518                     if((p%2)==0)
519                 glColor3f(1.0,0.816,0.0);
520                 else
521                     glColor3f(1.0,0.25,0.0);
522
523         glBegin(GL_POLYGON);//inner fume
524         glVertex2f(237.5,20+i);
525         glVertex2f(236.5,17.5+i);
526         glVertex2f(235.5,15+i);
527         glVertex2f(234.5,12.5+i);
528         glVertex2f(233.5,10+i);
529         glVertex2f(232.5,7.5+i);
530         glVertex2f(236,5+i);
531         glVertex2f(239.5,2.5+i);
532         glVertex2f(243,0+i);
533         glVertex2f(246.5,-2.5+i);
534         glVertex2f(250,-5+i);
535         glVertex2f(253.5,-2.5+i);
536         glVertex2f(257,0+i);
537         glVertex2f(260.5,2.5+i);
538         glVertex2f(264,5+i);
539         glVertex2f(267.5,7.5+i);
540         glVertex2f(266.5,10+i);
541         glVertex2f(265.5,12.5+i);
542         glVertex2f(264.5,15+i);
543         glVertex2f(263.5,17.5+i);
544         glVertex2f(262.5,20+i);//21 points
545
546         glEnd();
547         p=p+1;
548     for(j=0;j<=1000000;j++)
```

```
550         glutSwapBuffers();
551         glutPostRedisplay();
552         glFlush();
553     }
554
555     }
556
557     void rocket_in_motion()
558     {
559         count++;
560
561
562     for(i=195;i<=200;i++)
563     {
564          if(count>=5)
565          {
566                 glClearColor(0.0 ,0.0 ,0.0,1.0);
567             glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
568             if(flagl==0)
569             {
570             stars();
571             flagl=1;
572             }
573             else
574             {
575                 starsl();
576
577                 flagl=0;
578             }
579
580              }
581              else
582             {
583             glClearColor(0.196078  ,0.6 ,0.8,1.0);
584             glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
585              }
586             if(count>=100){
587             if(count<500){
588                mars(20.0*count*0.01);}
589                else(mars(20*5));
590              }
591             if(count<=130){
592             glColor3f(0.8,0.498039 ,0.196078);
593             glBegin(GL_POLYGON);//core
```

```
592         glColor3f(0.8,0.498039 ,0.196078);
593         glBegin(GL_POLYGON);//core
594             glVertex2f(237.5,20.0+i);
595             glVertex2f(262.5,20.0+i);
596             glVertex2f(262.5,120.0+i);
597             glVertex2f(237.5,120.0+i);
598         glEnd();
599         }
600
601         if(count>=150){
602             if(count>1000){
603
604             //Tesla
605             glColor3f(0.6667,0.6627,0.6784);
606             glBegin(GL_POLYGON);
607             glVertex2f(8.0+tx,250.0+tx);
608             glVertex2f(0.0+tx,258.0+tx);
609             glVertex2f(37.0+tx,290.0+tx);
610             glVertex2f(18.0+tx,290.0+tx);
611             glVertex2f(22.0+tx,293.0+tx);
612             glVertex2f(36.0+tx,294.0+tx);
613             glVertex2f(50.0+tx,309.0+tx);
614             glVertex2f(60.0+tx,300.0+tx);
615             glEnd();
616             drawFilledCircle(25.0+tx,262.0+tx,10.0);
617             drawFilledCircle(48.0+tx,285.0+tx,10.0);
618             tx+=0.04;
619
620
621
622         }
623             if(count>500){
624         if(int(count/100)%4!=0){
625         glColor3f(1.0,0.647,0.0);//satellite
626         glBegin(GL_POLYGON);//core
627             glVertex2f(237.5+yy,350.0-xx);
628             glVertex2f(252.5+yy,350.0-xx);
629             glVertex2f(252.5+yy,320.0-xx);
630             glVertex2f(237.5+yy,320.0-xx);
631         glEnd();
632         glColor3f(1.0,1.0,1.0);
633         glBegin(GL_POLYGON);//side-panels
634             glVertex2f(237.5+yy,340.0-xx);
635             glVertex2f(230+yy,340.0-xx);
```

```
634                 glVertex2f(237.5+yy,340.0-xx);
635                 glVertex2f(230+yy,340.0-xx);
636                 glVertex2f(230+yy,330.0-xx);
637                 glVertex2f(237.5+yy,330.0-xx);
638
639                 glVertex2f(262.5+yy,340.0-xx);
640                 glVertex2f(227.5+yy,340.0-xx);
641                 glVertex2f(227.5+yy,330.0-xx);
642                 glVertex2f(262.5+yy,330.0-xx);
643                 glEnd();
644                 if(xx>130){xx=130;yy=130;}
645                 else{
646                 xx+=0.1;
647                 yy+=0.1;}
648                 }
649                 else{xx=-40;
650                 yy=-40;}}
651                 else{
652             glColor3f(1.0,0.647,0.0);//satellite
653             glBegin(GL_POLYGON);//core
654                 glVertex2f(237.5,350.0);
655                 glVertex2f(252.5,350.0);
656                 glVertex2f(252.5,320.0);
657                 glVertex2f(237.5,320.0);
658             glEnd();
659             glColor3f(1.0,1.0,1.0);
660             glBegin(GL_POLYGON);//side-panels
661                 glVertex2f(237.5,340.0);
662                 glVertex2f(230,340.0);
663                 glVertex2f(230,330.0);
664                 glVertex2f(237.5,330.0);
665
666                 glVertex2f(262.5,340.0);
667                 glVertex2f(227.5,340.0);
668                 glVertex2f(227.5,330.0);
669                 glVertex2f(262.5,330.0);
670             glEnd();}
671             }
672
673             else{
674             glColor3f(1.0,1.0,1.0);//bonnet
675             glBegin(GL_POLYGON);//front
676             glVertex2f(237.5,120.0+i);
677             glVertex2f(262.5,120.0+i);
```

```
676         glVertex2f(237.5,120.0+i);
677         glVertex2f(262.5,120.0+i);
678         glVertex2f(250,170.0+i);
679         glEnd();
680         }
681
682         if(count<=120){
683         glColor3f(1.0,0.0,0.0);
684         glBegin(GL_POLYGON);//left_side_top
685         glVertex2f(237.5,120.0+i);
686         glVertex2f(217.5,95.0+i);
687         glVertex2f(237.5,95.0+i);
688         glEnd();
689             glBegin(GL_POLYGON);//left_side_bottom
690         glVertex2f(237.5,20.0+i);
691         glVertex2f(217.5,20.0+i);
692         glVertex2f(237.5,70.0+i);
693         glEnd();
694             glBegin(GL_POLYGON);//right_side_bottom
695         glVertex2f(262.5,20.0+i);
696         glVertex2f(282.5,20.0+i);
697         glVertex2f(262.5,70.0+i);
698         glEnd();
699             glBegin(GL_POLYGON);//right_side_top
700         glVertex2f(262.5,120.0+i);
701         glVertex2f(262.5,95.0+i);
702         glVertex2f(282.5,95.0+i);
703         glEnd();
704         }
705
706         if(count<=110){
707         glColor3f(0.556863 ,0.137255  ,0.419608);
708             glBegin(GL_POLYGON);//bottom_1_exhaust
709         glVertex2f(237.5,20.0+i);
710         glVertex2f(244.5,20.0+i);
711         glVertex2f(241,0.0+i);
712         glEnd();
713             glBegin(GL_POLYGON);//bottom_2_exhaust
714         glVertex2f(246.5,20.0+i);
715         glVertex2f(253.5,20.0+i);
716         glVertex2f(249.5,0.0+i);
717         glEnd();
718             glBegin(GL_POLYGON);//bottom_3_exhaust
719         glVertex2f(262.5,20.0+i);
```

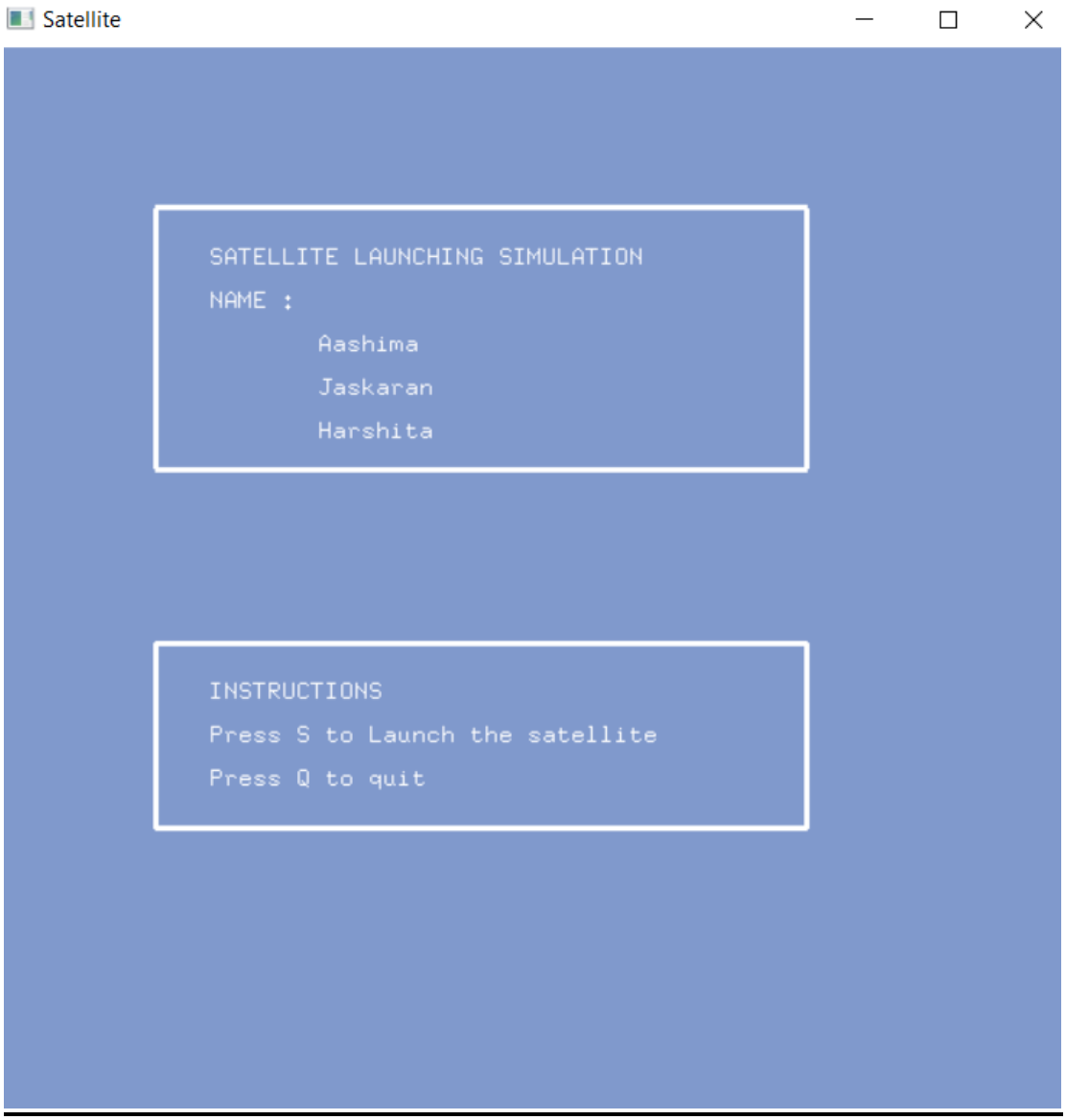
```
718             glBegin(GL_POLYGON);//bottom_3_exhaust
719         glVertex2f(262.5,20.0+i);
720         glVertex2f(255.5,20.0+i);
721         glVertex2f(258.5,0.0+i);
722         glEnd();
723         }
724
725     for(j=0;j<=1000000;j++)
726         ;
727     glutSwapBuffers();
728     glutPostRedisplay();
729     glFlush();
730 }
731 }
732
733 void mars(float radius)
734 {
735
736     glColor3f(1,0,0);
737    glBegin(GL_POLYGON);
738
739    for (int i=0; i<=359; i++)
740    {
741       float degInRad = i*DEG2RAD;
742       glVertex2f((300+f+cos(degInRad)*radius),(500-t+(sin(degInRad))*radius));
743    }
744
745    glEnd();
746    t=t+0.1;
747    if(t>200){
748     t=210;
749
750    }
751 }
752
753 //keys that trigger manual Lanch
754 void keyboard(unsigned char key, int x, int y)
755 {
756     if (key == 'S' || key == 's'){
757         for(int i=0;i<20000;i++)
758             static_rocket();
759         flag = 1;
760
761     }
```

```
760
761         }
762
763
764         if (key == 'Q' || key == 'q')
765             exit(0);
766
767     }
768
769     //design of homescreen
770     void page()
771     {
772         glColor3f(1, 1, 1);
773         glLineWidth(3);
774         glBegin(GL_LINE_LOOP);
775         glVertex2d(75, 425);
776         glVertex2d(375, 425);
777         glVertex2d(375, 305);
778         glVertex2d(75, 305);
779         glEnd();
780
781         drawstring(100, 400, "SATELLITE LAUNCHING SIMULATION");
782         drawstring(100, 380, "NAME : ");
783         drawstring(150, 360, "Aashima");
784         drawstring(150, 340, "Jaskaran");
785         drawstring(150, 320, "Harshita");
786
787         glBegin(GL_LINE_LOOP);
788         glVertex2d(75, 140);
789         glVertex2d(375, 140);
790         glVertex2d(375, 225);
791         glVertex2d(75, 225);
792         glEnd();
793
794         drawstring(100, 200, "INSTRUCTIONS");
795         drawstring(100, 180, "Press S to Launch the satellite");
796         drawstring(100, 160, "Press Q to quit");
797         glFlush();
798     }
799
800     //display all components
801     void display()
802     {
803         if (flag == 0)
```

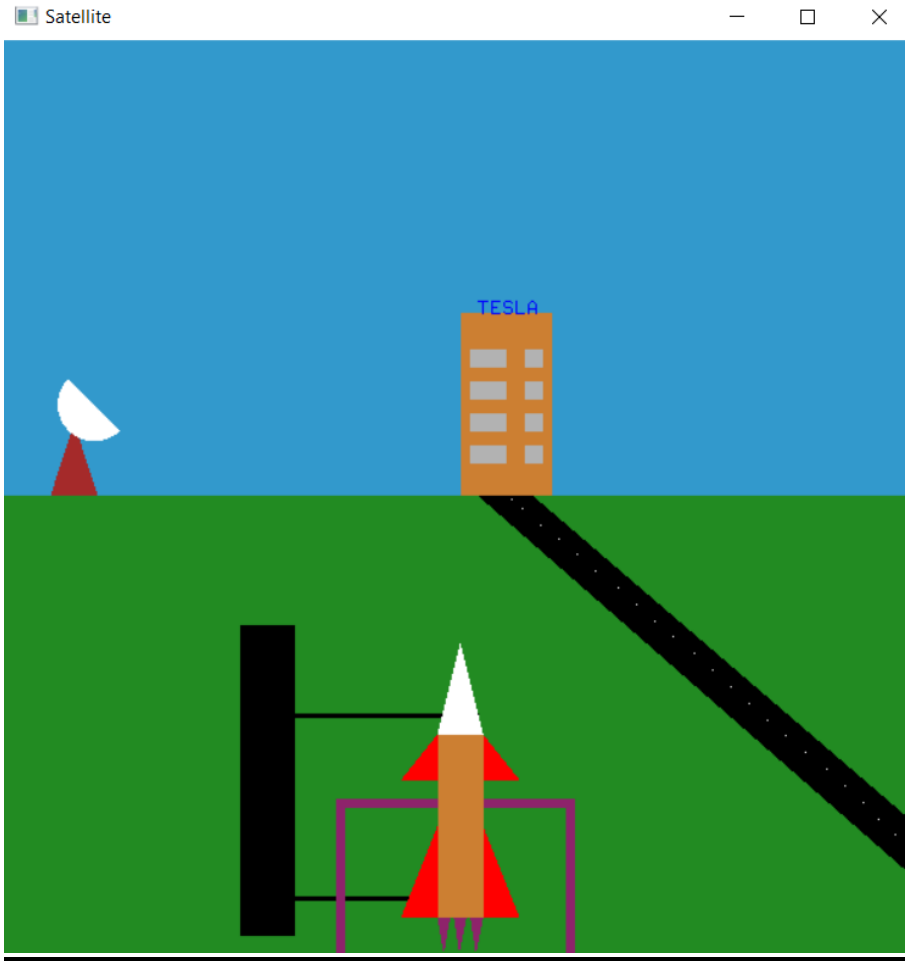
```
799
800   //display all components
801   void display()
802   {
803       if (flag == 0)
804       {
805           glClear(GL_COLOR_BUFFER_BIT);
806           page();
807           glutSwapBuffers();
808       }
809       else
810           control();
811       glFlush();
812   }
813
814
815   void myinit()
816   {
817       //int i;
818       glClearColor(0.5  ,0.6 ,0.8,1.0);
819
820
821       glPointSize(1.0);
822       gluOrtho2D(0.0,499.0,0.0,499.0);
823   }
824
825
826   int main(int argc,char*argv[])
827   {
828       glutInit(&argc,argv);
829       glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB);
830       glutInitWindowSize(600,600);
831       glutCreateWindow("Satellite");
832       myinit();
833       glutKeyboardFunc(keyboard);
834       glutDisplayFunc(display);
835       glutIdleFunc(display);
836
837
838
839      glutMainLoop();
840      return 0;
841   }
842
```

SCREENSHOTS:

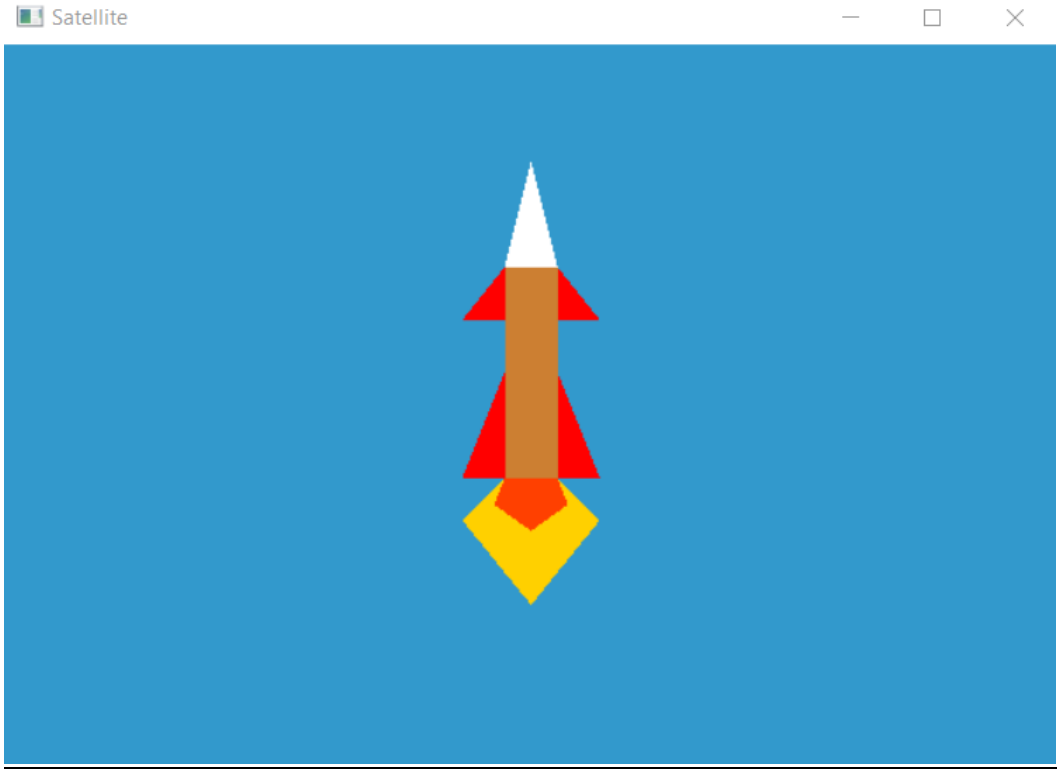
- Title Screen/Menu



- Initial Scene Before Launch:



- Atmosphere Flight Scene:



- Planet Scene with Tesla Cameo: